

Universitat de Lleida
Escola Politècnica Superior
Màster en Enginyeria de Programari Lliure

Treball de final de màster

Eina web de gestió de recollida de dades experimentals

Autor/a: Eulàlia Huguet Puig
Director: Carles Mateu Piñol
Octubre 2014

Índex de continguts

Introducció.....	6
Objectius.....	8
Tecnologies.....	12
Servidor Web: Apache.....	12
Bases de dades: Postgresql.....	12
Entorn de treball per al desenvolupament web: Django.....	13
Llenguatge de programació: Python.....	15
Llibreria de visualització de dades: D3.js.....	15
Història.....	16
Característiques tècniques.....	16
Exemple simple de funcionament.....	16
Altres eines i llenguatges de programació.....	17
Stack de programari: DjangoStack.....	18
Entorn de desenvolupament: Eclipse.....	19
Llenguatge: HTML.....	19
Llenguatge: CSS.....	19
Llenguatge: JavaScript.....	20
Llibreria: JQuery.....	20
Llibreria: Twitter Bootstrap.....	20
Mòdul de Python: Widget del calendari per a Bootstrap i Django.....	21
Aplicació.....	22
Metodologia i desenvolupament.....	22
Anàlisi de requeriments.....	23
Disseny.....	24
Arquitectura de la solució.....	24
Base de dades.....	25
Implementació.....	26
Fitxer de configuració: settings.py.....	26
El model: fitxer model.py.....	27
La vista: Fitxers del directori Template.....	28
El controlador: el fitxer views.py.....	29
L'administrador: admin.py.....	31
Llista de fitxers.....	31
Resultat.....	32
Llicències.....	43
Conclusions.....	45
Treball Futur.....	46
Bibliografia.....	47

Índex de taules

Taula 1: Descripció dels sensors del cubicle: Station.....	9
Taula 2: Descripció dels sensors del cubicle: Caseta110.....	9
Taula 3: Resum de llicències del projecte.....	43

Índex d'il·lustracions

Il·lustració 1: Instal·lació experimental a Puigverd de Lleida.....	6
Il·lustració 2: Distribució dels cubicles en la instal·lació.....	8
Il·lustració 3: Algunes característiques del software TCS-01-RS i del TC-Register.....	10
Il·lustració 4: Logotip de HTTPd Apache.....	12
Il·lustració 5: Logotip de PostgreSQL.....	12
Il·lustració 6: Logotip de Django.....	13
Il·lustració 7: Funcionament de Django.....	14
Il·lustració 8: Generacions del desenvolupament web.....	15
Il·lustració 9: Logotip de Python.....	15
Il·lustració 10: Logotip de D3 Data-Driven Documents.....	15
Il·lustració 11: Exemple de gràfica amb D3.....	17
Il·lustració 12: Logotip de Bitnami.....	18
Il·lustració 13: Logotip d'Eclipse.....	19
Il·lustració 14: Logotip d'HTML5.....	19
Il·lustració 15: Logotip de CSS3.....	19
Il·lustració 16: Logotip de JQuery.....	20
Il·lustració 17: Logotip de Bootstrap.....	20
Il·lustració 18: Plantilla Dashboard de Bootstrap.....	20
Il·lustració 19: Widget selector de dates.....	21
Il·lustració 20: Etapes del model iteratiu i incremental.....	22
Il·lustració 21: Escenari d'execució de l'aplicació.....	24
Il·lustració 22: Model Entitat Relació.....	25
Il·lustració 23: Llista de fitxers que conformen el projecte.....	31
Il·lustració 24: Pantalla inicial de l'eina web Experiments.....	32
Il·lustració 25: Pantalla que mostra el resultat de l'experiment.....	33
Il·lustració 26: Pantalla que mostra la llista de sensors.....	34
Il·lustració 27: Pantalla que mostra les dades d'un sensor.....	35
Il·lustració 28: Pantalla d'autenticació.....	36
Il·lustració 29: Pantalla d'importació de dades.....	37
Il·lustració 30: Pantalla amb error en la importació de dades.....	38
Il·lustració 31: Pantalla resultant d'importar dades.....	39
Il·lustració 32: Pantalla d'administració de l'aplicació.....	40
Il·lustració 33: Pantalla d'administració dels fitxers.....	41
Il·lustració 34: Pantalla de modificació dels experiments.....	42
Il·lustració 35: Logotip GPLv3.....	43

Agraïments

Aquest treball de final de màster ha estat possible gràcies al suport i la paciència de les persones que m'envolten. Cadascuna d'elles ha col·laborat per alliberar-me de les meves tasques diàries perquè jo pogués arribar avui aquí.

Moltes gràcies Joel, Pep, pare i mare.

Introducció



El grup de recerca GREA Innovació concurrent (<http://www.grea.udl.cat>) disposa d'una instal·lació on s'experimenta amb diferents sistemes constructius, diferents materials i sistemes d'aïllament per tal de buscar la màxima eficiència energètica en la construcció tant per aplicacions industrials com per l'habitatge.

Aquesta instal·lació està situada a Puigverd de Lleida i està formada per cubicles com els que es veuen a la il·lustració següent:



Il·lustració 1: Instal·lació experimental a Puigverd de Lleida

Cadascun dels cubicles està instrumentat amb sondes de temperatura a les parets, al sostre i al terra i alguns cubicles també en tenen a l'interior de les parets. D'altres, tenen sondes que mesuren la humitat, la pressió, la radiació solar o l'energia consumida. Les sondes prenen mostres cada 5 minuts que envien a l'ordinador central de la instal·lació.

Es disposa d'un software privatiu proporcionat per STEP SL, el TCS01, que recull i fa un històric de les dades rebudes. Cada una o dues setmanes, els investigadors, es connecten a través d'un software de control remot, el TeamViewer, a l'ordinador central de la instal·lació i extreuen les dades de l'històric.

Les dades tenen un format propietari, l'hdt, que no es pot llegir fora d'aquest entorn. Si més no, el programa permet convertir aquests fitxers en fitxers de format CSV (de l'anglès comma-separated values).

Per organitzar les dades recollides el grup té un llibre on cada full representa un cubicle. A cada full hi consta el nom que han donat a cada fitxer i les sondes que en formen part. També s'hi especifica el nom, la descripció, el rang i la precisió de cada sonda.

Les sondes recullen dades constantment però no totes les dades són importants per la investigació. Es programen experiments al llarg del temps, que també estan especificats en un document, depenent de les demandes i de la línia d'investigació que segueixi el grup. Els experiments poden ser Free Floating (sense condicions) o BC (amb bomba de calor).

Un experiment s'acostuma a fer en diferents cubicles a la vegada per tenir les mateixes condicions ambientals però amb diferents sistemes envolvents o diferents materials d'aïllament per després fer la comparativa i comprovar quin és el més eficient. Els experiments duren entre 1 i 2 setmanes.

Per obtenir els resultats es fa servir un full de càlcul. Donat l'experiment del que es vol obtenir el resultat, es busquen els fitxers de dades amb el nom de cada cubicle i les dates, es copien en un full de càlcul, s'aplica un factor corrector a les dades per compensar l'error de la calibració de la sonda i se'n fan les gràfiques per fer-ne una comparativa.

Objectius

Com hem vist en el capítol anterior, el procés per obtenir una gràfica, per exemple de la temperatura d'una sonda durant un període determinat és complexe i es fa del tot manual. Seguidament analitzarem el perquè de l'extracció de les dades i l'obtenció dels resultats es fa d'aquesta manera.

Actualment el camp d'experimentació disposa de 22 casetes amb un total de 477 sondes o sensors.



Il·lustració 2: Distribució dels cubicles en la instal·lació

En la il·lustració es veu com estan distribuïts els cubicles. N'hi ha que són dobles en altura, tenen dos pisos, i uns altres que són dobles en amplada. Els que tenen alguna o totes les parets de color verd són els que estan destinats a la investigació de l'arquitectura vegetada: hi ha plantes en les cobertes i les façanes.

Es disposa d'una estació meteorològica (el cubicle Station) que recull dades constantment de la temperatura exterior, la humitat exterior, la radiació solar i de la direcció i velocitat del vent. En la taula 1 es mostra quins sensors té el cubicle etiquetat com a "Station" i en la taula 2 es mostra la mateixa informació pel cubicle etiquetat com a "Caseta110".

Nom del sensor	Descripció del sensor	Rang	N. decimals
AMB_EXT	Temperatura exterior	-10 a 50	0,1
H.OUTSIDE	Humitat exterior	0 a 100	0,1
SOL_CONTRO	Radiació solar	-20 a 1500	0,1
WIND_DIR	Direcció del vent	0 a 360	0,1
WIND	Velocitat del vent	-5 a 50	0,1

Taula 1: Descripció dels sensors del cubicle: Station

Nom del sensor	Descripció del sensor	Rang	N. decimals
AMB_TEM110	Temperatura ambient interior	-10 a 50	0,1
NORTH_I110	Temperatura interior de la pared nord	-10 a 50	0,1
EAST_IN110	Temperatura interior de la pared est	-10 a 50	0,1
WEST_IN110	Temperatura interior de la pared oest	-10 a 50	0,1
CEIL_IN110	Temperatura interior del sostre	-10 a 50	0,1
FLOOR_I110	Temperatura interior del terra	-10 a 50	0,1
SOUTH_I110	Temperatura interior de la pared sud	-10 a 50	0,1
SOUTH_E110	Temperatura exterior de la pared sud	-10 a 50	0,1
HUMIDIT110	Humitat relativa interior	0 a 100	0,1
ENERGY110	Energia consumida	0 endavant	0,1
POTACT110	Potència activa mesurada al comptador	0 endavant	0,1
VOLT110	Voltatge mesurat al comptador	0 a 500	0,1
CURRENT110	Corrent mesurada al comptador	0 a 100	0,1
IN_FLU110	Flux de calor al sostre	-50 a 50	0,1

Taula 2: Descripció dels sensors del cubicle: Caseta110

El programari que gestiona la recollida de dades és un programari privatiu i tancat. A la imatge 10 es veuen algunes de les característiques més importants que ofereix.

El programari va llegint les dades i les acumula en un històric. Quan l'històric està ple, comença a sobre escriure el primer registre, de manera que si no es guarda o buida l'històric es perden dades. El grup ha definit una estratègia per no perdre dades. Cada 2 setmanes s'ha de buidar l'històric i guardar-lo. Mentre es fa aquest procés, que dura uns 10 minuts, els registres que envien els sensors es perden perquè l'històric no està disponible per guardar-los.

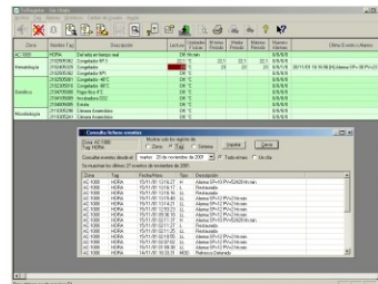
TCS-01-RS

- ❑ Registro de hasta 4000 variables (tags)
- ❑ Tablas de datos
- ❑ Temporizadores
- ❑ Listados de alarmas activadas
- ❑ Registros de tendencias en tiempo real
- ❑ Registros de tendencias históricos
- ❑ Sinópticos con animación
- ❑ Lógicas (And, Or, No, ExAnd; ExOr)
- ❑ Matemáticas: (+, -, *, /, raíz cuadrada)
- ❑ Compatible con cualquier regulador, PLC, data logger, etc, que tenga protocolo Modbus RTU



TC-REGISTER

- ❑ Registrador de medidas en continuo de un o varios equipos en Modbus RTU
- ❑ Dispone de 4 niveles de alarma por canal
- ❑ Registro de alarmas y de los cambios de consignas.
- ❑ Media, Máximo, Mínimo y Tiempo total en alarma.
- ❑ Tiempo de muestreo de 1 a 3600 muestras por hora
- ❑ Filtraje de las medidas



Il·lustració 3: Algunes característiques del software TCS-01-RS i del TC-Register

Aquest programari usa el seu propi format l'hdt per fer l'històric. Aquest format no es pot llegir directament, i es necessària l'exportació a un altre format conegut i estàndard de les dades, el CSV, per després poder tractar-les. L'exportació però també està limitada a 8 sondes per cada fitxer. Si un cubicle té més de 8 sensors s'han de fer dues o més exportacions, depenent del nombre de sensors. Quan fem una exportació (continuem amb el cubicle etiquetat com a "Caseta110") obtenim un fitxer com el següent (TEMP100.csv):

```
;"AMB_TEM110";"NORTH_I110";"EAST_IN110";"WEST_IN110";"CEIL_IN110";"FLOOR_I110";"SOUTH_I110";"SOUTH_E110"
09/10/2013 10:45:00; 18,9; 18,3; 18,2; 18,4; 19,1; 18,8; 18,4; 26
09/10/2013 10:50:00; 18,9; 18,3; 18,2; 18,4; 19,1; 18,8; 18,4; 27,2
09/10/2013 10:55:00; 18,9; 18,2; 18,2; 18,4; 19,1; 18,8; 18,4; 26,9
09/10/2013 11:00:00; 18,9; 18,2; 18,2; 18,3; 19,1; 18,8; 18,4; 28,5
09/10/2013 11:05:00; 18,9; 18,2; 18,2; 18,3; 19,1; 18,8; 18,3; 27,6
09/10/2013 11:10:00; 18,9; 18,2; 18,2; 18,3; 19,1; 18,8; 18,3; 29,3
09/10/2013 11:15:00; 18,9; 18,2; 18,2; 18,3; 19; 18,8; 18,3; 29,5
09/10/2013 11:20:00; 18,8; 18,2; 18,2; 18,3; 19,1; 18,7; 18,3; 30
09/10/2013 11:25:00; 18,8; 18,2; 18,2; 18,3; 19; 18,7; 18,3; 30,7
09/10/2013 11:30:00; 18,8; 18,2; 18,2; 18,2; 19; 18,7; 18,3; 31,8
09/10/2013 11:35:00; 18,8; 18,2; 18,2; 18,3; 19; 18,7; 18,3; 31,7
09/10/2013 11:40:00; 18,8; 18,2; 18,2; 18,2; 19; 18,7; 18,3; 32,4
.... (hem escurçat el fitxer només posant una mostra de les 13 primeres files)
```

Els fitxers estan formatats de la següent manera:

- La primera línia és la capçalera, està formada per una primera columna buida: la data i les següents columnes relacionen els sensors que formen part del cubicle, posats entre cometes dobles ("""). Els camps del fitxer estan delimitats per un punt i coma (;)
- Les altres línies són els valors que ha registrat cada sensor en una data concreta. La data segueix el format: dd/mm/aaaa hh:mm:ss i els nombres estan separats per comes: 18,4 (això representa 18.4, un nombre decimal). Cada línia conté en aquest exemple 8 mostres, el màxim que es pot exportar.

En total l'exemple té 12 files i 96 mostres ($12 * 8$).

Es fàcil veure el volum de dades que s'obté en tota la instal·lació: 1 sensor pren 1 mostra cada 5 minuts, 12 mostres cada hora, 288 mostres cada dia i 105.120 mostres cada any (suposant que no sigui un any de traspàs, no s'espatlli o marxi la llum per una tronada). Per 22 casetes que tenen 477 sensors tenim: **50.142.240** mostres en 1 any. Els històrics es treuen cada una o dues setmanes aproximadament, cada cop que es buiden els històrics es generen 66 fitxers diferents. Les dades mensuals ocupen unes 50Mb d'espai físic.

Quan l'investigador es planteja treure resultats d'aquestes dades, per exemple, l'evolució de la temperatura interior d'un cubicle durant el temps que ha durat l'experiment ha de treballar amb un full de càlcul format per dues columnes (com a mínim), la data i la temperatura i 2016 files (288 files per dia * 7 dies).

El cas habitual amb el que es troben és un experiment d'entre 1 i 2 setmanes que es fa a 3 cubicles a la vegada amb 10 o més sensors per cubicle, es a dir, un full de càlcul amb unes 30 columnes i entre 2000 o 4000 files, només de dades. Els altres fulls contenen les gràfiques de cada sensor que ha participat en l'experiment i algunes d'especials, per exemple:

- Una que compara l'energia gastada en cada cubicle diàriament i una altra l'acumulada
- Una que compara de la temperatura ambient interior que hi ha a dins de cada cubicle amb la de la temperatura exterior registrada per l'"Station".

I es clar! Tots, altre cop amb el factor de calibració aplicat.

No sembla gens fàcil treballar amb aquest volum de dades i tenir-les disponibles per consulta en qualsevol moment. Es gairebé impossible poder comparar mostres entre experiments. També es fa difícil fer càlculs amb les dades, que formen part de fitxers diferents.

L'objectiu del treball és dissenyar una eina que permeti als investigadors organitzar les dades i treure'n resultats de manera ràpida i senzilla.

A grans trets per aconseguir-lo hem de:

- Dissenyar, crear i posar en marxa una base de dades que ens permeti emmagatzemar de manera segura totes aquestes dades.
- Dissenyar i implementar una aplicació web que permeti:
 - importar les dades lliures d'error de format i d'estructura
 - gestionar els fitxers de dades
 - donar d'alta experiments i assignar-los els cubicles on es fan
 - visualitzar en una gràfica l'evolució d'un sensor en un període determinat
 - visualitzar en una gràfica el resultat d'un experiment

Tecnologies

Segons els objectius hem de crear una base de dades i hem d'implementar una eina web. Ens es necessari un gestor de bases de dades i un servidor web, per començar. Escollim DJANGO com a entorn de treball per al desenvolupament d'aplicacions web, httpd Apache com a servidor web i PostgreSQL com a sistema gestor de base de dades.

Servidor Web: Apache



Il·lustració 4: Logotip de HTTPd Apache

Apache HTTP Server és un servidor HTTP de codi obert per a sistemes operatius moderns, incloent UNIX i Windows. L'objectiu d'aquest projecte és proporcionar un servidor segur, eficient i extensible que proporciona els serveis d'HTTP en sincronització amb els estàndards HTTP actuals. Apache httpd és un dels servidors web més populars a Internet. L'Apache HTTP Server ("httpd") és un projecte de la Fundació Apache Software.

<http://httpd.apache.org/>

Bases de dades: PostgreSQL



Il·lustració 5: Logotip de PostgreSQL

PostgreSQL és un potent sistema de base de dades relacional, de codi obert. Compta amb més de 15 anys de desenvolupament actiu i una arquitectura provada que s'ha guanyat una sòlida reputació de fiabilitat, integritat de dades i correcció. Funciona en tots els principals sistemes operatius, incloent Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), i Windows. És totalment compatible amb ACID

, suporta claus foranes, combinacions, vistes, triggers i procediments emmagatzemats (en diversos idiomes). Inclou a més de SQL: 2008 tipus de dades, incloent INTEGER, numèrics, booleans, CHAR, VARCHAR, DATE, INTERVAL, i TIMESTAMP. També és compatible amb l'emmagatzematge d'objectes binaris grans, com imatges, sons o vídeo. Compta amb interfícies de programació natives per a C / C++, Java, NET, Perl, Python, Ruby, Tcl, ODBC, entre d'altres. <http://www.postgresql.org/about/>

L'aplicació web que hem de dissenyar té dos paràmetres clau: el volum de dades que ha de suportar (50.000.000 de registres a l'any) i les consultes que poden fer els usuaris de l'aplicació que en aquest moment són 3. En un futur podran augmentar el nombre de cubicles en 2 més en els propers 4 anys i el personal dedicat a aquesta investigació en concret es mantindrà igual o augmentarà en 1 persona més en els pròxims anys.

No es preveu doncs un augment significatiu que ens faci pensar en un canvi de tecnologia referent a la bases de dades. Sí que serà necessari que disposem d'emmagatzemament suficient i determinen com accedirem a les dades.

Podem fer una ullada a les bases de dades que usen empreses reconegudes mundialment, Twitter i Facebook per a consultar les seves dades amb nombres massius i extraordinaris en quantitat de transaccions i usuaris:

Volum que suporta MySQL de Twitter:

- Més de 140 milions d'usuaris actius
- 4629 tweets per segon (que poden arribar a 25,000 en hores punta)
- 3 milions de registres generats cada dia
- 400 milions de tweets cada dia, replicats 4 vegades.

Volum que suporta MySQL de Facebook (2011):

- Més de 950 milions d'usuaris actius
- 450 milions de registres llegits per segons en hores punta
- Consultes per segon: 13 milions (en hora punta)
- Temps de resposta de consultes: 4ms per lectura, 5ms per escriptura.
- Registres modificats per segon: 3,5 milions (en hores punta)

Dades extretes de l'article escrit per Brett Winterford on Oct 5, 2012 7:00 AM - Twitter, PayPal reveal database performance

Read more: http://www.itnews.com.au/News/317811_twitter-paypal-reveal-database-performance.aspx#ixzz3GDMKis7y

Evidentment no només usen aquesta tecnologia de bases de dades. Twitter usa una llarga llista de tecnologies: MySQL, FlockDB, Memcache, Cassandra, Gizzard, Lucense, Hbase/Hadoop i Redis i Facebook usa MySQL, Memcache, HAYSTACK, Hadoop i Cassandra entre d'altres. Aplicat al nostre cas, amb Postgresql en tenim prou per emmagatzemar les dades i gestionar-les sense cap altra ajuda.

Entorn de treball per al desenvolupament web: Django

Django és un entorn de treball per al desenvolupament d'aplicacions web de codi obert escrit totalment en Python que fomenta la creació de llocs web complexos en poc temps i amb un disseny net. Django promou la reusabilitat de codi, la connectivitat de components, el desenvolupament ràpid i usa el principi de “No et repeteixis” (DRY, de l'anglès, Don't Repeat Yourself). <https://www.djangoproject.com/>



*Il·lustració 6:
Logotip de
Django*

Aquesta eina de desenvolupament pertany a la tercera generació de entorns de treball de desenvolupament web (Il·lustració 8) i permet seguir metodologies de desenvolupament de programari evolutiu i iteratiu.

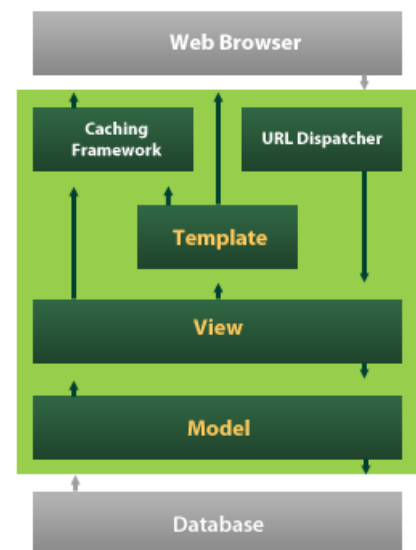
Algunes de les seves principals característiques són:

- Un mapejador objecte-relacional (ORM)
- Aplicacions que es poden instal·lar a una pàgina que sigui gestionada amb Django

- Una API de base de dades robusta
- Incorpora una un sistema de “vistes genèriques” que fan les tasques més habituals de la programació web.
- Té un sistema de plantilles basat en etiquetes, amb herència a les plantilles.
- Un “dispatcher” de URL's basat en expressions regulars
- Un sistema “middleware” per a desenvolupar característiques addicionals. La distribució de Django per defecte inclou components de middlewere que proporcionen caché, normalització de URL's, protecció CSRF i suport a sessions d'usuari.
- Suport a la internacionalització.

Django funciona de la següent manera:

1. El navegador (Web browser) envia una sol·licitud
2. El URL Dispatcher interpreta la sol·licitud i tria la vista apropiada que l'ha de processar
3. La vista (View) interactúa amb el model i aquest a la vegada amb la base de dades per obtenir dades si fa falta
4. La vista crida a la plantilla
5. La plantilla renderitza la resposta a la sol·licitud demana per navegador

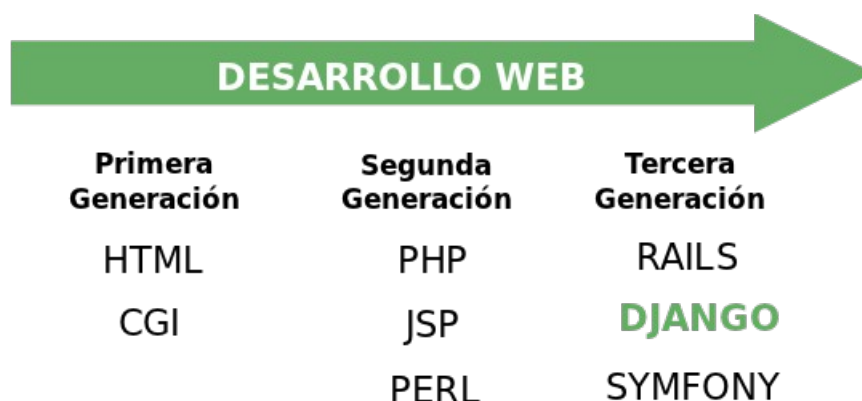


Il·lustració 7: Funcionament de Django

Django està inspirat en l'arquitectura de desenvolupament Model Vista Controlador (MVC) encara que no la compleix estrictament, el que s'anomena controlador en el model estricte el patró MVC, Django l'anomena Vista i al que s'anomenaria Vista, Django l'anomena Template. (MVT)

Respecte a les bases de dades, Django suporta PostgreSQL, MySQL i SQLite3, la recomanada és PostgreSQL. Un cop s'han creat els models Django proporciona una abstracció de la base de dades mitjançant l'API que té definida que ens permet crear, consultar, actualitzar i esborrar objectes. També permet executar consultes SQL directament. En el model de dades de Django, una classe representa una taula a la base de dades i un objecte de la classe un registre de la taula.

Django porta per defecte un servidor web molt senzill per fer proves en la fase de desenvolupament que només suporta un usuari connectat a la vegada.. A la fase de producció es recomana fer servir Apache2 amb el mod_python. També pot funcionar sobre altres servidors web com Lighttpd.



Il·lustració 8: Generacions del desenvolupament web

Llenguatge de programació: Python



Il·lustració 9: Logotip de Python

Python és un llenguatge de programació d'alt nivell. La seva filosofia de disseny busca llegibilitat en el codi i la seva sintaxi permet als programadors expressar conceptes en menys línies de codi del que seria possible en llenguatges com C. Python suporta diversos paradigmes de programació, incloent-hi programació orientada a objectes, imperativa i

també funcional o procedimental. Presenta un sistema dinàmic i una gestió de la memòria automàtica i té una gran i exhaustiva biblioteca estàndard. <http://www.python.org>

Libreria de visualització de dades: D3.js



Il·lustració 10: Logotip de D3 Data-Driven Documents

D3.js (o simplement D3 Data-Driven Documents) és una biblioteca JavaScript per generar gràfics SVG (Scalable Vector Graphics) a partir de dades. Utilitza estàndards com el HTML5 i el CSS3 i permet un gran control sobre el resultat. Farem servir aquesta llibreria per a generar les gràfiques dels experiments i les gràfiques de l'evolució dels sensors en un període determinat. Library released under BSD license. Copyright 2013 Mike Bostock. <http://d3js.org/>

Història

Posant-nos en situació, els navegadors web en els seus inicis només eren capaços de mostrar pàgines web estàtiques. S'han buscat moltes solucions per superar aquesta limitació. Una de les més significatives va ser la integració del JavaScript com a llenguatge d'script als navegadors. Gradualment el JavaScript ha esdevingut el llenguatge estàndard per crear pàgines web en les que l'usuari pugui interactuar.

Paral·lelament s'han buscat eines per poder presentar dades dins de les pàgines web. Dels molt projectes que s'han fet els més notables han estat **Prefuse**(2005), fa servir Java i les presentacions de dades es rendejaven al navegador amb el plug-in de Java i **Flare** (2007), que usa ActionScript i necessita el plug-in de Flash.

El 2009 i en base a l'experiència adquirida en l'ús del Prefuse i el Flare, el Professor Jeff Heer, l'estudiant de doctorat Mike Bostock, i l'estudiant de màster Vadim Ogievetsky del grup “Stanford Visualization Group” de la Universitat d'Stanford van crear Protovis, una llibreria de JavaScript per generar gràfiques SVG a partir de dades. Aquesta llibreria va tenir molt bona acceptació tant des del món acadèmic com des del món professional.

El 2011, el desenvolupament de Protovis es va aturar per centrar-se en un nou projecte: el D3.js. De les suggerències rebudes amb Protovis, Bostock, juntament amb Heer and Ogievetsky van desenvolupar el D3.js per proporcionar un marc més expressiu, que a la vegada, es centres amb els estàndards i proporcionés un millor rendiment.

Característiques tècniques

Va incrustat en una pàgina HTML i utilitza funcions predefinides de JavaScript per seleccionar elements, crear objectes SVG, estils, afegir transicions o efectes dinàmics(events). Als objectes els podem aplicar els estils CSS. Les dades es vinculen als objectes SVG usant funcions simples de la llibreria. Es poden generar textos, gràfiques i diagrames. Les dades poden estar en diversos formats, els més comuns: JSON, CSV i GeoJSON, però si es necessari es poden escriure funcions de JavaScript que reconguin altres formats de dades.

Exemple simple de funcionament

```
<!DOCTYPE html>
<meta charset="utf-8">
<style>

.chart div {
  font: 10px sans-serif;
  background-color: steelblue;
  text-align: right;
  padding: 3px;
  margin: 1px;
  color: white;
}

</style>
```



```
<div class="chart"></div>
<script src="http://d3js.org/d3.v3.min.js"></script>

<script>

var data = [4, 8, 15, 16, 23, 42];

var x = d3.scale.linear()
    .domain([0, d3.max(data)])
    .range([0, 420]);

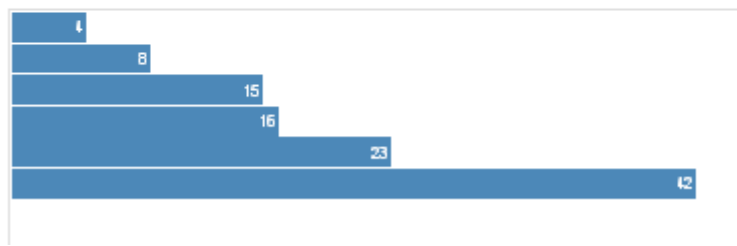
d3.select(".chart")
    .selectAll("div")
    .data(data)
    .enter().append("div")
    .style("width", function(d) { return x(d) + "px"; })
    .text(function(d) { return d; });

</script>
```

Aquest codi HTML més la llibreria D3.js i l'script genera la següent gràfica:

mbostock's block #7322386 November 5, 2013

Bar Chart I



Il·lustració 11: Exemple de gràfica amb D3

Exemple obtingut de: <http://bl.ocks.org/mbostock/7322386>

Aquestes aplicacions i llibreries conformem el nucli amb el que s'executarà l'eina web un cop acabada. Per al desenvolupament ens hem d'ajudar d'altres eines i llenguatges de programació.

Altres eines i llenguatges de programació

Una de les coses que sempre porta més temps a l'hora de fer funcionar una aplicació web, encara que sigui en desenvolupament, és la correcta configuració de la base de dades, el servidor web i l'entorn d'execució.

Hem d'instal·lar l'Apache i el mod_python i configurar-lo per a que faci de servidor web, hem d'instal·lar i configurar el Postgres i hem de configurar el Django i lligar-lo a la base de dades que crearem. Per solventar aquest inconvenient, ens hem ajudat d'un stack.

Un stack és un conjunt de programes que treballen conjuntament per a produir un resultat o qualsevol grup d'utilitats o aplicacions de rutina que funcionen com un conjunt. Arxius instal·lables, definicions de programari i pegats poden ser inclosos en un conjunt de programari. Un stack dels més populars basat en Linux és LAMP (Linux, Apache, MySQL, Perl o PHP o Python) o el WINS (Windows Server, Internet Explorer, NET, SQL Server) és un stack de programari basat en Windows.

Stack de programari: DjangoStack



*Il·lustració 12:
Logotip de Bitnami*

Bitnami és botiga d'aplicacions de servidors i entorns de desenvolupament que es pot instal·lar a l'ordinador portàtil, en una màquina virtual o directament en el núvol. Bitnami s'encarrega de la compilació i configuració de les aplicacions i totes les seves dependències: llibreries de tercers, llenguatge i les bases de dades. El programari empaquetat resultant és un 'stack' que es posa a disposició com instal·ladors nadius o com una màquina virtual.

https://bitnami.com/learn_more

Bitnami té una llarga llista d'stacks preparats per fer servir tant en Linux, com en Mac com en Windows. Alguns dels que hi ha disponibles són: l'stack de Joomla, el de Moodle, Drupal, WordPress, Ruby, JRuby, Tomcat, etc. Per fer el desenvolupament de tota la eina web, farem servir l'stack de Django: **DjangoStack** que porta incorporat el següent programari:

- Django 1.5.5: is released under the BSD license, which is located <http://code.djangoproject.com/browser/django/trunk/LICENSE>
- Python 2.7.6: is released under the Python License, wich is located at <http://www.python.org/download/releases/2.7.3/license/>
- SQLite 3070603: is released under the Public Domain license, which is located at http://en.wikipedia.org/wiki/Public_Domain
- MySQL 5.5.34: is distributed under the GNU General Public License v2, which is located at <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>
- Apache 2.4.7:is distributed under the Apache License v2.0, which is located at <http://www.apache.org/licenses/LICENSE-2.0>
- PostgreSQL 9.3.1-1: is distributed under the BSD License, which is located at <http://www.freebsd.org/copyright/license.html>
- Boto AWS API 2.6.0
- OpenSSL is released under the terms of the Apache License, which is located at <http://www.openssl.org/source/license.html>

- Zlib is released under the zlib License (a free software license/compatible with GPL), which is located at http://www.gzip.org/zlib/zlib_license.html
- Libiconv is released under the LGPL license, which is located at <http://www.gnu.org/licenses/lgpl.html>
- The Python Imaging Library (PIL) is Copyright © 1997-2011 by Secret Labs AB, Copyright © 1995-2011 by Fredrik Lundh. More details in <http://www.pythonware.com/products/pil/license.htm>
- Freetype is released under The Freetype Project License, that is located at <http://freetype.sourceforge.net/FTL.TXT>

Entorn de desenvolupament: Eclipse

Eclipse és un entorn integrat de desenvolupament de codi obert programat principalment en Java (per tant, multiplataforma), per a desenvolupar projectes en C, C++, COBOL, Python, Perl, PHP, i molts altres, sempre quan s'instal·lin els connectors corresponents per a cada llenguatge de programació. Farem servir Eclipse Standard/SDK - Version: Kepler Service Release 2 amb el PyDev for Eclipse. Eclipse Public License - v 1.0 <http://www.pydev.org/>



Il·lustració 13: Logotip d'Eclipse

L'eclipse ens servirà per a fer tot el codi de l'aplicació, des dels mòduls de python fins al codi html de la pàgina web.



Il·lustració 14: Logotip d'HTML5

Llenguatge: HTML

HTML (acrònim d'Hyper Text Markup Language, en català, "llenguatge de marcat d'hipertext"), és un llenguatge de marcat que deriva de l'SGML dissenyat per estructurar textos i relacionar-los en forma d'hipertext. Gràcies a Internet i als navegadors web, s'ha convertit en un dels formats més populars que existeixen per a la construcció de documents per a la web. <http://www.w3.org/html/>

Aquest llenguatge ens permetrà escriure el codi de la interfície web.

Llenguatge: CSS

CSS (Cascading Style Sheets, en català Fulls d'Estil en Cascada) és un llenguatge de fulls d'estil utilitzat per descriure la semàntica de presentació (l'aspecte i format) d'un document escrit en un llenguatge de marques. La seva aplicació més comuna és dissenyar pàgines web escrites en HTML i XHTML, però el llenguatge també pot ser aplicat a qualsevol classe de document XML, incloent-hi SVG i XUL. CSS està dissenyat principalment per permetre la separació de contingut del document (escrit en HTML o un llenguatge de marques similar) de la presentació del document, incloent-hi elements com la disposició, colors, i fonts.



Il·lustració 15: Logotip de CSS3

Llenguatge: JavaScript

JavaScript (abreviat com a "JS") és un llenguatge de programació interpretat. S'usa principalment en el costat del client, implementat com a part d'un navegador web permetent millores a la interfície d'usuari i fent que les pàgines web siguin dinàmiques.

Llibreria: JQuery

jQuery és una biblioteca o framework de JavaScript que permet simplificar la manera d'interaccionar amb els documents HTML, manipular l'arbre DOM, gestionar esdeveniments, desenvolupar animacions i afegir interacció amb la tecnologia AJAX en pàgines web. jQuery ofereix una sèrie de funcionalitats basades en Javascript que d'una altra manera requerien molt més codi, és a dir, amb les funcions d'aquesta biblioteca s'aconsegueixen bons resultats en menys temps i espai.



Il·lustració 16: Logotip de JQuery

Llibreria: Twitter Bootstrap

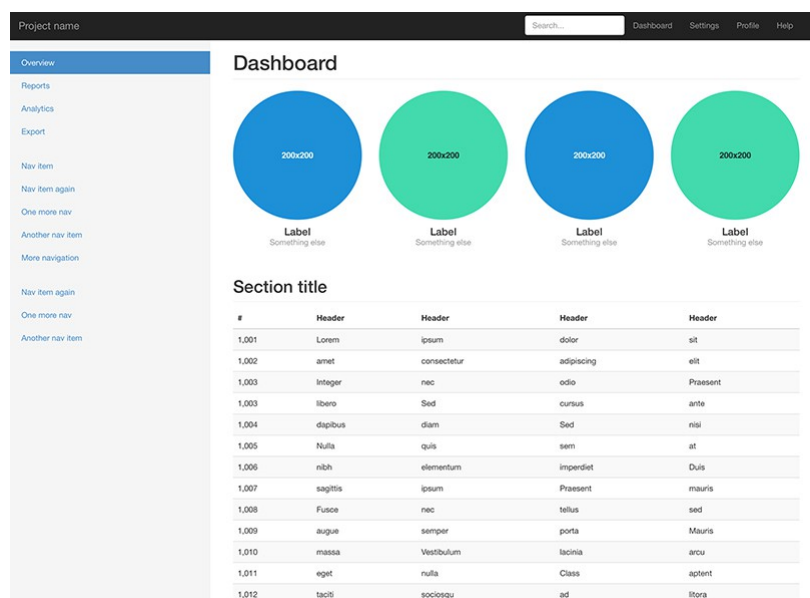


Il·lustració 17: Logotip de Bootstrap

Twitter Bootstrap (versió 3) és un framework o conjunt d'eines de software lliure per dissenyar llocs i aplicacions web. Conté plantilles de disseny amb una tipografia determinada a més de formularis, botons, quadres, menús de navegació i d'altres elements propis de disseny, tot basat en HTML i CSS. També disposa d'extensions de JavaScript opcionals. Code licensed under MIT, documentation under CC BY 3.0

<http://getbootstrap.com/>

La plantilla que més s'adapta al disseny de la interfície web que necessitem és la "Dashboard". Té una barra a la part superior per posar les accions de l'usuari (iniciar/tancar sessió, anar a la part d'administració) una part de menú a l'esquerra per les opcions de l'aplicació (Experiments, Importar dades, Sensors) i en la part central podem presentar les dades de l'aplicació.



Il·lustració 18: Plantilla Dashboard de Bootstrap

Mòdul de Python: Widget del calendari per a Bootstrap i Django

En previsió que serà necessari fer consultes pel camp “data”, el formulari web que necessiti demanar aquest camp hauria d'estar pensat per fer fàcil a l'usuari la introducció de les dates i que tinguin el format adequat de dia, mes, any, hora minut i segon.

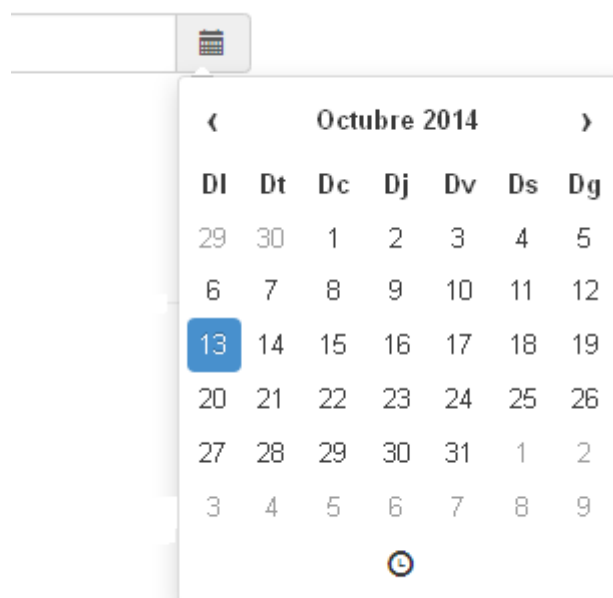
Al repositori de paquets de Python tenim el paquet:django-bootstrap3-datetimepicker-2.2.3 <https://pypi.python.org/pypi/django-bootstrap3-datetimepicker>

L'ha fet Nakahara Kunihiro i té llicència: License: Apache License 2.0.

La pàgina del projecte és: <https://github.com/nkunihiro/django-bootstrap3-datetimepicker>

Per instal·lar-lo hem de fer servir l'instal·lador de paquets de python i afegir l'aplicació a django:

- Executar: `pip install django-bootstrap3-datetimepicker`
- Afegir 'bootstrap3_datetime' a les nostres INSTALLED_APPS (django)



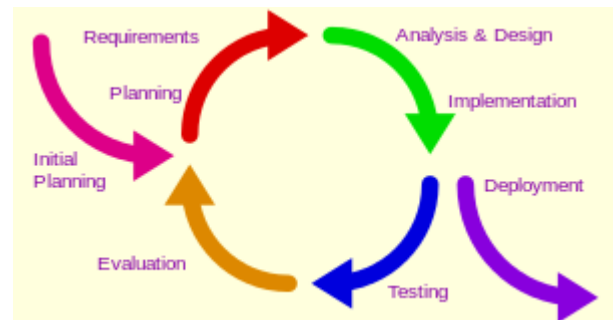
Il·lustració 19: Widget selector de dates

Aplicació

En aquest capítol, el més llarg, ens centrarem en explicar com hem desenvolupat el projecte, quines decisions de disseny hem pres, com hem fet la implementació i quins resultats hem obtingut.

Metodologia i desenvolupament

Hem escollit un *model de desenvolupament iteratiu i incremental* per al desenvolupament del programari. Un projecte es planifica en diversos blocs temporals, que s'anomenen iteracions. Cada iteració es pot entendre com un "projecte en miniatura", s'ha de fer l'anàlisi de requeriments, el disseny, la implementació, les proves i la documentació.



Il·lustració 20: Etapes del model iteratiu i incremental

Per a cada iteració, el procés de treball és similar, de manera que el client pot anar obtenint els beneficis del producte de manera incremental. El producte evoluciona, a partir dels resultats obtinguts de les iteracions anteriors, perquè s'hi afegixen nous objectius i/o requeriments o perquè es milloren els que ja existien. Fins a l'etapa final que es fa el desplegament. Un aspecte fonamental en aquest model de desenvolupament és la prioritització dels objectius o requeriments en funció dels valor que aporten a l'usuari.

Aquest tipus de model de desenvolupament permet que els requeriments no estiguin del tot especificats al començament del projecte cosa que pot provocar que el programari no estigui ben estructurat i sigui difícil de mantenir però es pot adaptar al canvi de requeriments més fàcilment.

Fent ús d'aquest procés de desenvolupament obtenim els següents avantatges en contra dels model tradicional de desenvolupament:

- Els usuaris no han d'esperar a l'etapa final per a fer ús del projecte, des de la primera iteració tenen un producte que compleix els requeriments principals.
- Retroalimentació amb l'usuari a cada iteració per a modificar, afegir o prioritzar requeriments que necessita.
- Podem resoldre problemes de risc en iteracions inicials de desenvolupament i corregir-los, o corregir-los en la iteració següent
- Tenim la visió que el projecte avança des del primer moment
- Els projectes desenvolupats sota aquest mètode tenen menys risc de fracàs, millor productivitat de l'equip i tenen una menys quantitat de defectes
- L'aprenentatge i l'experiència de cada iteració permet augmentar la productivitat i d'aquesta manera ajustar la planificació del projecte.

No se li coneixen grans desavantatges al mètode però sí que hem de ser curosos perquè el sol fet de fer servir el mètode no dóna cap garantia d'èxit al projecte

Anàlisi de requeriments

L'usuari ens ha proporcionat una llista de coses que li agradaria que fes l'aplicació:

1. Una taula per cada caseta.
2. Hi haurà gràfics predefinits que es criden amb un botó.
3. Es podran fer gràfics nous totalment flexibles.
4. Hi haurà alarmes i controls per verificar dades exportades.
5. Es podran fer cerques de dades concretes per valors, data, ¿experiment?
6. Es podran fer cerques per creuar dades i fer gràfics creuats de diferents variables en el mateix moment temporal o de la mateixa variable en diferents moments temporals.
7. Hi hauria d'haver un índex on s'especifiqués el programa d'experiments realitzat i on es poguessin mostrar només les setmanes amb un tipus concret d'experiments.
8. Els gràfics han de tenir un format adequat per a la seva utilització en informes i articles. Els eixos s'han d'autoajustar. Tot i això hi ha d'haver flexibilitat per canviar títols, mida i tipus de lletra, colors, etc.
9. Els gràfics han de poder tenir eix y principal i secundari on mostrar diferents variables.
10. Cada variable ha de tenir un color assignat en funció de la representació a fer.
11. S'han de poder visualitzar per pantalla diferents gràfics al mateix temps.
12. S'ha de poder ampliar el sistema creant més taules fàcilment per poder gestionar més casetes.
13. S'han de poder fer operacions amb les dades abans de fer els gràfics.
14. Els gràfics han de mostrar automàticament informació sobre el tipus d'experiment que s'està mostrant i si hi ha alguna particularitat durant aquell període (errors, sensors trencats, etc.).
15. Els gràfics s'han de poder guardar en format jpg, tif i excel.
16. S'han de poder fer mitges horàries, diàries, etc. de les diferents dades.

Si agrupem els requeriments que parlin del mateix tema, ens podem fer una idea de les parts que ha de tenir l'aplicació i com han d'estar distribuïdes:

- gestió: 1,12
- cerca: 5,6
- planificació: 7
- dades: 4,16
- gràfiques: 2,3,6,8,9,10,11,13,14,15

Es veu clarament que l'usuari incideix repetidament en el tema de les gràfiques. Suposem que la manera que té de treballar fins ara (el full de càlcul), fa que tendeixi a demanar tot el que li fa falta, per exemple el requeriment 11, més d'una gràfica per pantalla, i no vol perdre cap de les funcionalitats de les que disposa: requeriment 3, es podran fer gràfiques noves i totalment flexibles.

A més també busca millora amb el tema de la cerca sobre les dades per poder fer estadística sobre els seus experiments.

Un cop hem ordenat els requeriments i hem estructurat tota la informació rebuda, proposem, en aquesta etapa inicial de planificació els següents requeriments:

1. Els usuaris accediran a l'aplicació mitjançant un login i una contrasenya.
2. Algunes parts de l'aplicació han d'estar protegides i només hi podran accedir els usuaris autenticats i en altres parts no caldrà autenticació.
3. L'aplicació ha de tenir una interfície d'administració per mantenir els usuaris, les casetes, els sensors, els experiments i les dades recollides. (R1,R12)
4. L'aplicació ha de permetre afegir les dades dels experiments i donar informació sobre errors en les dades.(R4)
5. L'aplicació ha de mostrar els resultat d'un experiment: Mostrant les dades bàsiques de l'experiment i una gràfica que mostri les mesures que han recollit els sensors. (R2,R8,R9,R10,R11,R14,R15,R16)
6. L'aplicació ha de permetre consultar les mesures que ha recollit un sensor durant una període de temps.(R5)

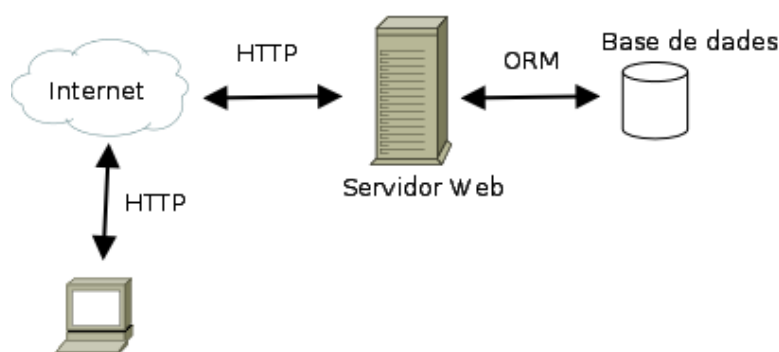
Hi ha alguns requeriments que no posarem en aquesta iteració inicial i els deixarem per a les següents iteracions:

- R3: gràfiques noves i totalment flexibles
- R6: cerques de dades creuades i la fer gràfica
- R7: planificador d'experiments
- R13: càlculs sobre les dades: per exemple aplicar la calibració de la sonda a les dades.

Disseny

Arquitectura de la solució

L'estructura del sistema és la que es veu a la figura següent:



Il·lustració 21: Escenari d'execució de l'aplicació

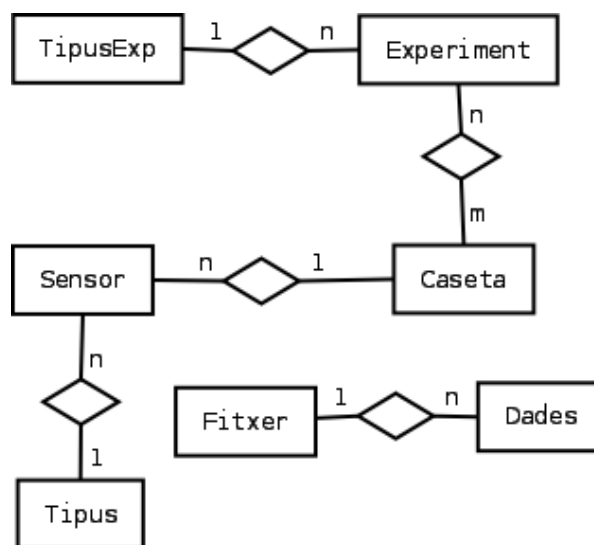
L'usuari es connectarà al sistema mitjançant un navegador web i a través d'Internet accedirà als serveis que li ofereix l'aplicació.

Usarem el patró d'arquitectura de software model-vista-controlador (MVC) ja que necessitem separar les dades, la lògica de l'aplicació i la interfície d'usuari. Tot i que en un principi aquest patró es va desenvolupar per a aplicacions d'escriptori, s'ha adaptat àmpliament com l'arquitectura per a dissenyar i implementar aplicacions web.

Actualment hi ha molts frameworks que implemenen aquest patró, per exemple Symfony amb el llenguatge de programació PHP, Django amb Phyton o Spring amb Java. Tots ells només es diferencien en la interpretació de com es divideixen les funcions del patró MVC entre el client i el servidor.

Base de dades

A la il·lustració 22 hem representat el model entitat relació per a la implementació de la persistència de dades de l'aplicació.



Il·lustració 22: Model Entitat Relació

El model entitat relació el traduïm al model relacional de la següent manera:

TipusExp(id, nom, descripcio)
Experiment(id, nom, descripcio, data_inici, data_fi, id_TipusExp)
Caseta(id, nom, descripcio)
CasetesExpRel(id_Experiment, id_Caseta)
Sensor(id, nom, descripcio, id_Tipus, id_Caseta)
Tipus(id, nom, rang_min, rang_max, precisio)
Fitxer(id, nom, importat, data_fi, data_inici)
Dades(data, variable, valor, id_Fitxer)

L'entitat Dades és l'entitat que representa el valor d'un sensor (camp variable) en un moment del temps (camp data). Aquest valor ha de ser únic i només pot pertànyer a un únic fitxer de dades.

Implementació

Fitxer de configuració: settings.py

Com ja hem dit en capítols anteriors farem la implementació de l'aplicació amb el framework Django. Django posa en el fitxer de configuració `settings.py` la configuració d'accés a la base de dades, si fem servir internacionalització, quines aplicacions tenim instal·lades en el nostre site, etc.

A continuació posem algunes de les línies més destacades d'aquest fitxer:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'projecte',          # Or path to database file if using sqlite3.
        # The following settings are not used with sqlite3:
        'USER': 'postgres',
        'PASSWORD': 'xxxxxxx',
        'HOST': '',                 # Empty for localhost
        'PORT': '',                 # Set to empty string for default.
    }
}

# Llenguatge de l'aplicació
LANGUAGE_CODE = 'ca-ES'
# Internacionalització activada
USE_I18N = True
# URL on es penjaran els recursos, css, js, fitxers
MEDIA_URL = '/media/'

# Aplicacions instal·lades, per defecte, hem activat la part d'administració
# i les 2 que hem inclòs pel projecte
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    # Uncomment the next line to enable the admin:
    'django.contrib.admin',
    # Aplicacions que fem servir o hem creat pel projecte
    'experiments',
    'bootstrap3_datetime',
)

MIDDLEWARE_CLASSES = (
    'django.middleware.common.CommonMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    # Uncomment the next line for simple clickjacking protection:
    # 'django.middleware.clickjacking.XFrameOptionsMiddleware',
)
```

El model: fitxer model.py

Un cop hem configurat les opcions i característiques de l'aplicació, hem de definir el model, i ho fem al fitxer *model.py*. Hem d'escriure el model relacional anterior amb la sintaxi pròpia del framework. Per exemple:

Tipus(id, nom, rang_min, rang_max, precisió), s'escriuria de la següent manera:

```
class Tipus(models.Model):
    nom = models.CharField(max_length=50)
    descripcio = models.CharField("descripció", max_length=100)
    rmin = models.IntegerField("rang mínim")
    rmax = models.IntegerField("rang màxim")
    precisió = models.IntegerField("precisió")

    def __unicode__(self):
        return self.nom
```

Si no definim cap clau primària, Django afegirà un atribut més a aquesta classe: **id** i crearà una seqüència de nombres enters positius per a que sigui clau primària en la base de dades.

Tenim un model relacional senzill, tot i això Django 1.5.5 no suporta la gestió multicolumna de la clau primària, no podem posar dos columnes com a clau primària, només en podem definir una. Això ens afecta a dues taules: CasetesExpRel i Dades

En el cas de CasetesExpRel, és la taula que es crea de la relació N-M, Django farà el següent:

CasetesExpRel(id, id_Experiment, id_Caseta)

```
class CasetaExpRel(models.Model):
    experiment = models.ForeignKey(Experiment)
    caseta = models.ForeignKey(Caseta)
```

Aquest canvi no impacta en el disseny. No és el cas de la taula Dades, Django farà el següent:

Dades(id, data, variable, valor, id_Fitxer)

```
class Dades(models.Model):
    fitxer = models.ForeignKey(Fitxer)
    data = models.DateTimeField()
    variable = models.CharField(max_length=50) #nom sensor
    valor = models.DecimalField(decimal_places=1, max_digits=12) # valor sensor
```

Hem solventat aquest imprevist fent la gestió de la inserció de duplicats a nivell de programari, cada cop que s'importa un fitxer de dades i es fa un INSERT a la taula Dades s'ha de fer la comprovació de duplicats.

La vista: Fitxers del directori Template

Un cop tenim el model, definim la vista, (el que django anomena Template). Hem fet servir el sistema de plantilles de Django amb herència. Hem definit una plantilla que s'anomena **base.html** en la que es basaran totes les pàgines de l'aplicació, que té la següent estructura (hem exclòs les línies no importants):

```
<!DOCTYPE html>
<html lang="ca-ES">
<head>
    <!-- Línies de la capçalera -->
    {% block extrahead %}{% endblock %}
</head>
<body>
{% load tags %}
<!-- Barra de navegació superior -->
<div class="container-fluid">
<div class="row">
    <div class="col-sm-3 col-md-2 sidebar">
        <!-- Barra de navegació lateral esquerra -->
    </div>
    <div class="col-sm-9 col-sm-offset-3 col-md-10 col-md-offset-2 main">
        <!-- Bloc de continguts -->
        {% block content %}{% endblock %}
        <footer>
            <br /><br />
            <center><p> &copy Open project 2014 </p></center>
        </footer>
    </div>
</div> <!-- row -->
</div> <!-- container-fluid -->
</body>
</html>
```

Tots els altres pàgines del directori templates tenen la següent estructura:

```
{% extends 'base.html' %}

{% block title %} {% endblock %}

{% block content %}

{% endblock %}
```

Entre les etiquetes de `{% block title %} {% endblock %}` posarem el títol i entre les etiquetes de `{% block content %} {% endblock %}` posarem la presentació de la corresponent pàgina. D'aquesta manera si volem modificar l'estructura de l'aplicació i posar el menú de navegació a la dreta en lloc de l'esquerra, només hem de modificar la plantilla base.html.

El controlador: el fitxer `views.py`

Per últim només ens falta fer la part del controlador, el que django anomena vista. Al fitxer `views.py` hem de definir les vistes de l'aplicació. Una vista típica:

```
def about(request):  
    return render_to_response('about.html', context_instance=RequestContext(request))
```

La vista `about`, rep una petició `request`, i retorna com a resposta fer un render del template “`about.html`”.

La vista **`importar_dades`** és la que s'encarrega de fer la importació de les dades. Llegim el fitxer CSV dues vegades:

- una en memòria per comprovar errors i la duplicitat de les dades,
- i la segona, si no ha donat cap problema la primera, per guardar les dades a la base de dades.

S'ha fet d'aquesta manera perquè s'ha de garantir que tot el fitxer de dades sigui correcte, i això no ho podem garantir fins que arribem a la lectura de la última mostra. Per tant hem de fer una transacció, normalment d'unes 2500 mostres, es a dir 2500 inserts a la base de dades. En cas d'error ho hauríem de desfer fent un rollback.

Durant el procés d'importació el fitxer CSV hem de tenir en compte que llegim d'un fitxer de text pla. Tot el que obtindrem seran cadenes de text o String en Python. Demanem a l'usuari que ens digui quin és el delimitador de les columnes i quin és el separador dels números.

Per obtenir la data que llegim del fitxer (`row[0]`) com una cadena de text “09/10/2013 10:45:00” fem:

```
data_inici = timezone.datetime.strptime(row[0], '%d/%m/%Y %H:%M:%S')
```

Per obtenir un número decimal de la cadena de text que llegim: “18,3” fem:

```
if 'separador' != '.':  
    cadena = cadena.replace(separador, ".")
```

L'altre procés important és el que genera la gràfica. Hi ha dos vistes que fan servir la funció de generació i funcionen de la mateixa manera. Com hem vist en capítols anteriors usem la llibreria D3.js que incrusta un objecte `<script></script>` a la part del client. La vista ha de generar el codi javascript i retornar-lo a la plantilla per a que el navegador el pugui interpretar. Cridem a la funció de la següent manera:

```
cadena = generar_script(cadena, sensor.nom, data_inici, data_fi)
```

Al acabar la funció `generar_script`, `cadena` conté tot el codi necessari per dibuixar la gràfica.

La funció `generar_script`:

```
def generar_script(cadena, sensor, data_inici, data_fi):

    cadena+='<script>
    // Set the dimensions of the canvas / graph
    var margin = {top: 30, right: 20, bottom: 30, left: 50},
        width = 600 - margin.left - margin.right,
        height = 270 - margin.top - margin.bottom;
    // .... línies suprimides
    var svg = d3.select("#'")
    cadena+=sensor
    cadena+='")
        .append("svg")
            .attr("width", width + margin.left + margin.right)
            .attr("height", height + margin.top + margin.bottom)
        .append("g")
            .attr("transform",
                "translate(" + margin.left + "," + margin.top + ")");

    // Get the data
    '''

    cadena+=obtenir_dades(sensor, data_inici, data_fi)
    cadena+='
        data.forEach(function(d) {
            d.date = parseDate(d.date);
            d.close = +d.close;

        });
    // .... línies suprimides
    </script>
    '''

    return cadena
```

I la vista retorna:

```
def singleSensor(request, id_sensor):
    sensor = Sensor.objects.get(id=id_sensor)
    cadena=''
    if request.method == 'POST':
        form = Escollir2DatesForm(request.POST)
        if form.is_valid():
            data_inici = form.cleaned_data['data_inici']
            data_fi = form.cleaned_data['data_fi']
            cadena+='<div id='+sensor.nom+'></div>'
            cadena = generar_script(cadena, sensor.nom, data_inici, data_fi)
    else: #GET
        form = Escollir2DatesForm()

    ctx = {'sensor': sensor, 'form': form, 'str': mark_safe(cadena)}
    return render_to_response('SingleSensor.html', ctx,
    context_instance=RequestContext(request))
```

`cadena` conté tot el codi html i javascript necessari. Però per poder-la enviar com a codi html i no com a String, hem de fer servir `mark_safe`. Per recollir la variable a la plantilla només hem de posar: `{{ str }}`.

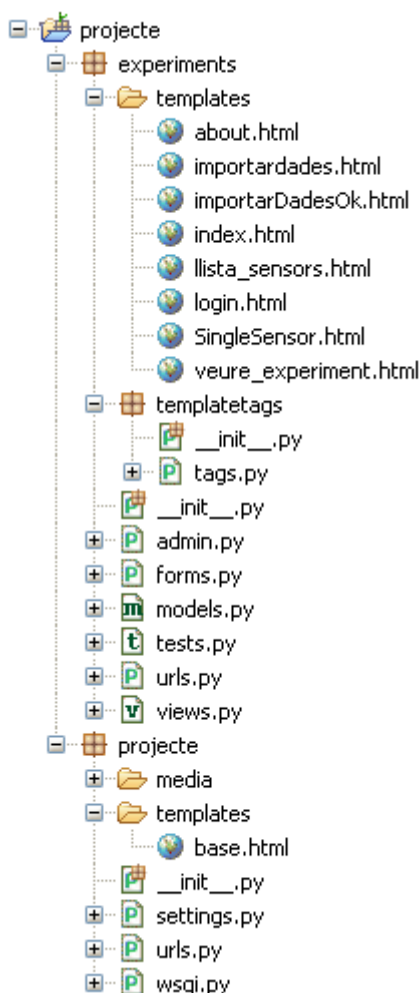
L'administrador: admin.py

Per la part de l'administració, Django ens proporciona un fitxer anomenat *admin.py* on podem definir quins models hi han d'aparèixer i quin serà el seu comportament:

```
class FitxerAdmin(admin.ModelAdmin):
    search_fields = ['nom']
    readonly_fields=('importat','data_inici', 'data_fi',)
    list_display = ('nom', 'importat','data_inici', 'data_fi')

admin.site.register(Fitxer, FitxerAdmin)
```

Hem definit la classe *FitxerAdmin*, amb un camp de cerca per nom (*search_fields*). Que ens mostri els camps de: nom, importat, data_inici, data_fi (*list_display*) i que aquestes tres últims no es puguin modificar des de la part d'administració (*readonly_files*). I per últim registrem la classe *Fitxer* que pertany al model i la classe *FitxerAdmin* que defineix el comportament al site admin.



Llista de fitxers

La il·lustració 23 mostra la distribució dels paquets i dels fitxers que intervenen en el projecte.

El paquet *projecte*, conté dos directoris:

- El directori *media*, conté tots els fitxers estàtics, com són els css, les llibreries js, el directori de dades on s'allotjen tots els fitxers csv que afegim
- i el directori *templates* on hi ha la plantilla per a la resta de l'aplicació

i els fitxers de *settings.py*

El paquet *experiments*, conté el model de dades: *models.py* (model), el directori *templates* amb les pàgines web (*view*) i el de les vistes: *views.py* (controller)

Il·lustració 23: Llista de fitxers que conformen el projecte

Resultat

Seleccioneu una data:

2011-01-24 12:33

Enviar

Llista d'experiments

A partir de: 24 de gener de 2011 a les 12:33

Nom	Descripció
Experiment 1 -- del 4 de juny de 2012 a les 10:15 a 11 de juny de 2012 a les 10:15	Exp 1 - BC22
Experiment 2 -- del 13 de juny de 2012 a les 9:00 a 18 de juny de 2012 a les 9:00	Exp 2 - BC27
Experiment 3 -- del 25 de juny de 2012 a les 9:00 a 1 de juliol de 2012 a les 9:00	Exp 3 - BC29
Experiment 4 -- del 2 de juliol de 2012 a les 9:00 a 12 de juliol de 2012 a les 9:00	Exp 4 - BC24
Experiment 5 -- del 12 de juliol de 2012 a les 9:00 a 21 de juliol de 2012 a les 9:00	Exp 5 - BC18
Experiment 6 -- del 21 de juliol de 2012 a les 16:00 a 1 de agost de 2012 a les 16:00	Exp 6 - FF

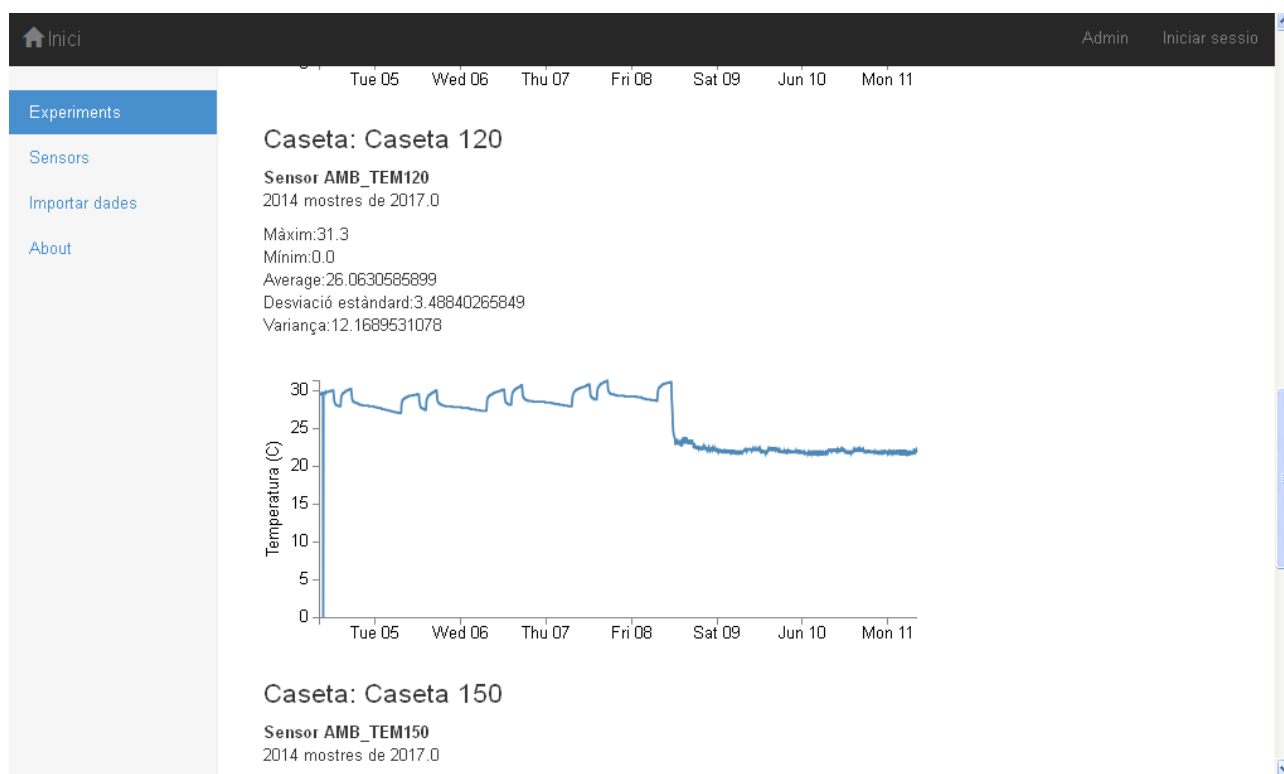
© Open project 2014

Il·lustració 24: Pantalla inicial de l'eina web Experiments

Aquesta és la pàgina principal de l'aplicació. Es mostra una llista d'experiments a partir de la data actual. Si volem consultar altres experiments anteriors, seleccionarem la data en la caixa de calendari i farem clic a “Enviar”. El sistema ens retornarà la llista d'experiments a partir d'una data escollida.

Fent clic al nom de l'experiment podrem veure els resultats obtinguts.

L'usuari accedeix a aquesta la pàgina principal de l'eina i sense necessitat d'autenticació.



Il·lustració 25: Pantalla que mostra el resultat de l'experiment

Al accedir a un experiment, obtenim les dades de l'experiment:

Descripció: Exp 1 - BC22

Data inici: 4 de juny de 2012 a les 10:15

Data final: 11 de juny de 2012 a les 10:15

Casetes que participen en l'experiment:

Caseta 110

Caseta 120

Caseta 150

i una gràfica per a cada sensor de temperatura ambient: AMB_TEMPxxx, on xxx és el número de la caseta al que pertany, i de la caseta "Station". Per una caseta en concret es mostra:

Caseta: Caseta 120

Sensor AMB_TEM120

2014 mostres de 2017.0

Màxim: 31.3

Mínim: 0.0

Average: 26.0630585899

Desviació estàndard: 3.48840265849

Variança: 12.1689531078

Inici
Admin
Iniciar sessió

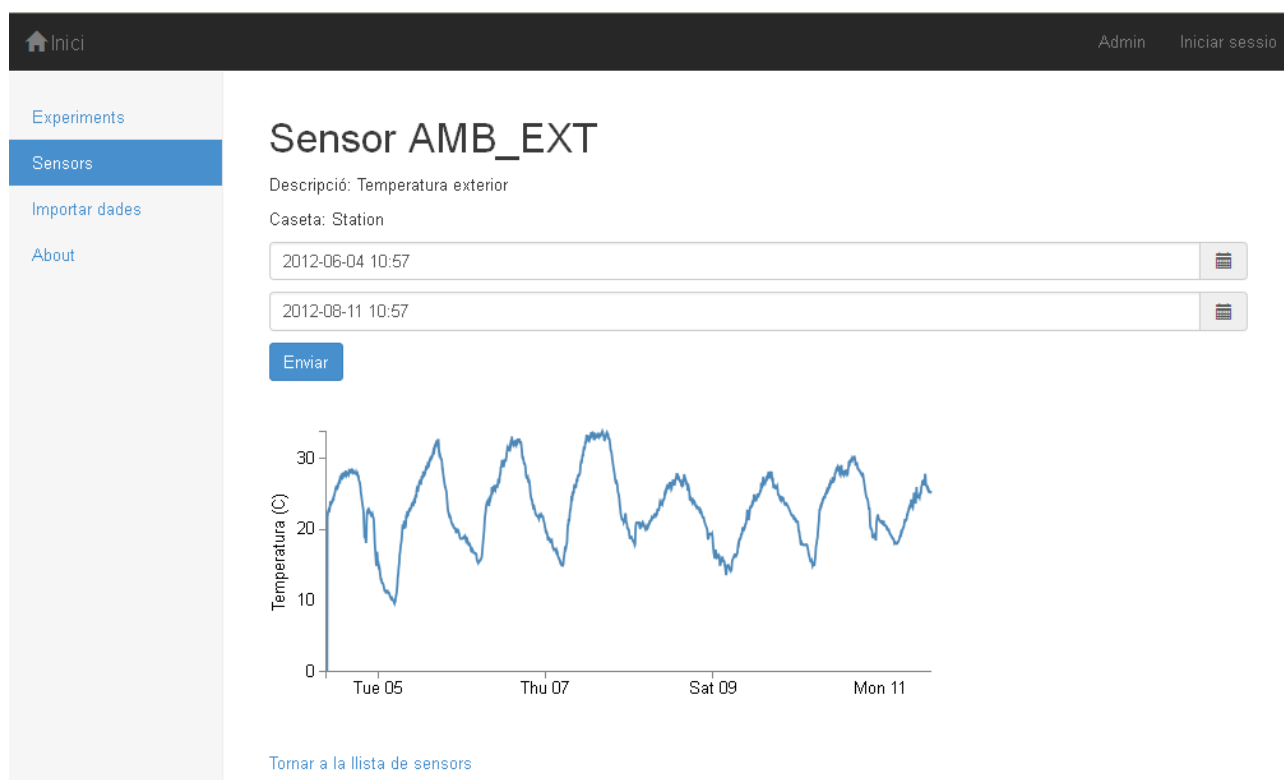
Experiments
Sensors
Importar dades
About

Llista de sensors

Nom	Descripció	Caseta	
AMB_EXT	Temperatura exterior	Station	Veure
H.OUTSIDE	Humitat exterior	Station	Veure
SOL_CONTRO	Radiació solar	Station	Veure
WIND_DIR	Direcció del vent	Station	Veure
WIND	Velocitat del vent	Station	Veure
AMB_TEM10	Temperatura ambient interior a 1,2m	Caseta 10	Veure
HUMIDIT10	Humitat relativa interior	Caseta 10	Veure
CEIL_FS_10	Temperatura sostre entre el formigó i suro	Caseta 10	Veure
CEIL_SG_10	Temperatura sostre entre el suro i guix	Caseta 10	Veure
NORTH_FS10	Temperatura entre formigó i suro de la pared nord	Caseta 10	Veure

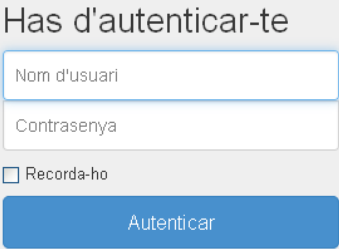
Il·lustració 26: Pantalla que mostra la llista de sensors

Amb l'enllaç “Sensors” es poden llistar els sensors que actualment estan a la instal·lació experimental, fent clic al nom o al botó veure, si tenen dades registrades, podrem veure la seva gràfica.



Il·lustració 27: Pantalla que mostra les dades d'un sensor

Pàgina resultant de la consulta el sensor AMB_EXT de la caseta "Station" des del 04-06-2012 a les 10:57 fins el 11-06-2012 a les 10:57.



Has d'autenticar-te

Nom d'usuari

Contrasenya

☐ Recorda-ho

Autenticar

Il·lustració 28: Pantalla d'autenticació

Pàgina d'autenticació. Els usuaris es validen a la base de dades local de django.

Importar dades

Delimitador dels camps de les dades: ;

Separador de decimals: ,

Interval entre les mostres (en minuts): 5

Caseta: -----

Fitxer que no tenen les dades importades: -----

Enviar Netejar

© Open project 2014

Il·lustració 29: Pantalla d'importació de dades

A la pàgina per importar dades es necessita autenticació. Hem d'omplir un formulari amb:

- Delimitador dels camps: quin caràcter ens separa les columnes del fitxer CSV, per defecte un punt i coma “;”
- Separador de decimals: per defecte una coma (,)
- Interval entre mostres: per defecte 5minuts
- Hem de seleccionar dels desplegable, el fitxer de dades que volem importar i a quina caseta pertany (només a efectes de comprovació de la capçalera del fitxer i els sensors que pertocuen)

[Inici](#) [Benvingut/da eulalia](#) [Admin](#) [Tancar sessió](#)

Benvingut/da eulalia

Última connexió: 12 de octubre de 2014 a les 12:32

[Experiments](#)

[Sensors](#)

[Importar dades](#)

[About](#)

Importar dades

Delimitador dels camps de les dades: ;

Separador de decimals: ,

Interval entre les mostres (en minuts): 5

Caseta: Caseta 110

Fitxer que no tenen les dades importades: dades/2012/07/24/OTHE110.CSV

Enviar

Netejar

Info: Hi ha errors al fitxer

Error: Les dates del fitxer no són correlatives

Inici: 2012-07-12 13:10:00

Data: 2012-07-13 23:55:00

Fila: ['13/07/2012 00:00:00', '60,8', '7573', '17170432', '805307072', '-520193', '-14']

© Open project 2014

Il·lustració 30: Pantalla amb error en la importació de dades

Un cop hem seleccionat els paràmetres fem “Enviar”. En aquesta captura s'ha produït un error al fitxer. El procés s'atura i mostra la línia del fitxer que ha provocat l'error (en vermell):

Fitxer OTHER110.CSV

```
13/07/2012 23:45:00; 61,6; 7525; 17170432; 805307072; -520193; -14
13/07/2012 23:50:00; 69; 7533; 17170432; 805307072; -520193; -11
13/07/2012 23:55:00; 61,3; 7556; 17170432; 805307072; -520193; -13
13/07/2012 00:00:00; 60,8; 7573; 17170432; 805307072; -520193; -14
14/07/2012 00:00:00; 60,8; 7573; 17170432; 805307072; -520193; -14
14/07/2012 00:05:00; 62,9; 7584; 17170432; 805307072; -520193; -13
14/07/2012 00:10:00; 62; 7602; 17170432; 805307072; -520193; -12
```

Aquest error es produeix sovint, les dates de la fila següent amb la fila anterior no són més grans o iguals a 5 minuts. L'usuari hauria d'aplicar una acció correctiva en aquest fitxer i tornar a llançar el procés d'importació.

The screenshot shows the 'Importació de dades' page. On the left is a sidebar with navigation links: 'Inici', 'Benvingut/da eulalia', 'Experiments', 'Sensors', 'Importar dades' (highlighted), and 'About'. The main content area displays the following information:

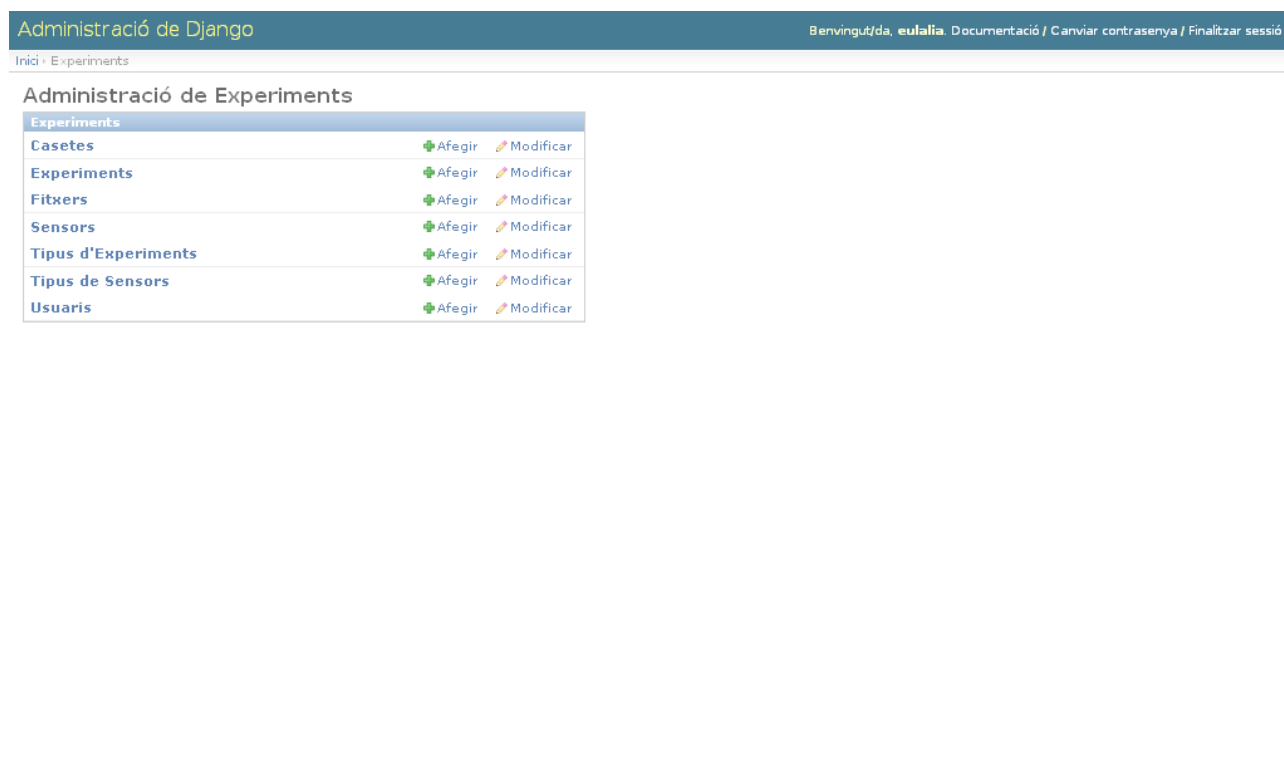
- Info:** Dades guardades correctament ✓
- Files:** Tractades 2633 de 3215,0 possibles
- Nombre de sensors:** 8
- Mostres:** Importades 21064 de 25720,0 possibles
- Data_inici:** 12 de juliol de 2012 a les 13:10
- Data_fi:** 23 de juliol de 2012 a les 17:00
- Links: [Importar més dades](#) | [Anar a l'inici](#)
- Footer: © Open project 2014

Il·lustració 31: Pantalla resultant d'importar dades

Si el procés no té cap error finalitza i mostra un resum del nombre de dades importades. En l'exemple veiem que s'han tractat 2633 files de les 3215 possibles (mostres de 5minuts en 5 minuts entre 12-07-2012 a les 13:10 fins al 23-07-2012 a les 17:00). Si revisem el fitxer TEMP110.CSV que hem importat i trobem que a les darreres files falten dades:

```
21/07/2012 18:15:00; 17,7; 19,3; 19,3; 19,6; 17,7; 19,9; 18,8; 35,2
21/07/2012 18:20:00; 18,1; 19,3; 19,4; 19,6; 17,8; 19,8; 19; 34,7
21/07/2012 18:25:00; 17,6; 19,3; 19,2; 19,6; 17,7; 19,8; 18,8; 34,6
21/07/2012 18:30:00; 17,4; 19,2; 19,2; 19,6; 17,6; 19,8; 18,8; 33,1
23/07/2012 17:00:00; 0; 0; 0; 0; 0; 0; 0; 0
```

Des del dia 21-07-2012 a les 18:30 no s'ha registrat cap dada, fins el 23-07-2012 a les 17:00 i malauradament són totes zero. És possible que tot i que el fitxer és correcte i el procés ha resultat correcte l'investigador hauria de descartar aquestes dades i tornar a repetir l'experiment.



Il·lustració 32: Pantalla d'administració de l'aplicació

Django ens proporciona una part completa d'administració del model, amb el seu estil i format predeterminat que no hem modificat per tenir una visió diferent a la de l'aplicació. Des del punt de vista de l'usuari li és més senzill reconèixer que està a la interfície d'administració si veu un entorn diferent al de l'aplicació.

Hem registrat totes les classes per poder inserir, modificar o eliminar registres, excepte la classe Dades. La classe Dades, només es pot modificar fent la importació del fitxer corresponent.

Administració de Django Benvingut/da, eulalia. Documentació / Canviar contrasenya / Finalitzar sessió

Inici > Experiments > Fitxers

Seleccioneu fitxer per modificar Afegir fitxer +

Q Cerca

Acció: Anar 0 de 56 seleccionats

<input type="checkbox"/>	Nom	Importat	Data inici	Data fi
<input type="checkbox"/>	dades/2012/08/02/STATION.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/08/02/PCM_150.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/08/02/OTHE150.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/08/02/OTHE110.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/08/02/OTHE120.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/08/02/TEMP150.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/08/02/TEMP120.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/08/02/TEMP110.CSV	●	23 de juliol de 2012 a les 17:15	1 de agost de 2012 a les 13:35
<input type="checkbox"/>	dades/2012/07/24/STATION.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/07/24/PCM_150.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/07/24/OTHE150.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/07/24/OTHE120.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/07/24/OTHE110.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/07/24/TEMP150.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/07/24/TEMP120.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/07/24/TEMP110.CSV	●	12 de juliol de 2012 a les 13:10	23 de juliol de 2012 a les 17:00
<input type="checkbox"/>	dades/2012/07/14/STATION.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/07/14/PCM_150.CSV	○	(Cap)	(Cap)
<input type="checkbox"/>	dades/2012/07/14/OTHE150.CSV	○	(Cap)	(Cap)

Il·lustració 33: Pantalla d'administració dels fitxers

En aquesta part només afegim els fitxers de dades. Al inserir un fitxer, fem una còpia al servidor en el directori “dades”, on estaran tots els fitxers de dades. Com tots tenen el mateix nom, perquè identifiquen un grup de sensors que pertanyen a una caseta, els posarem en subdirectoris de l'any, el mes i el dia referents a la data en que es va inserir a la base de dades.

El camp “Importat” per defecte és “False”. I en la interfície d'administració és un camp de només lectura, l'usuari no el pot canviar. Només es posa a “True” si la importació del fitxer s'ha fet correctament. També són de lectura els camps de data_inici i data_fi que també s'omplen quan es llegeix el fitxer de dades.

Administració de Django
Benvingut/da, eulalia. Documentació / Canviar contrasenya / Finalitzar sessió ió

Inici » Experiments » Experiments » Experiment 1 - Exp 1 - BC22

Modificar experiment
Històric

Nom: Experiment 1

Descripció: Exp 1 - BC22

Típus: BC22

Data inici: Data: 04/06/201: Avui | Hora: 10:15:00 Ara |

Data fi: Data: 11/06/201: Avui | Hora: 10:15:00 Ara |

Casetes que intervenen en l'experiment

Caseta	Eliminar?
Experiment 1 - Exp 1 - BC22 - Caseta 110 Caseta 110	<input type="checkbox"/>
Experiment 1 - Exp 1 - BC22 - Caseta 120 Caseta 120	<input type="checkbox"/>
Experiment 1 - Exp 1 - BC22 - Caseta 150 Caseta 150	<input type="checkbox"/>

Afegir un/a altre/a Caseta.

Eliminar
Desar i afegir-ne un de nou
Desar i continuar editant
Desar

Il·lustració 34: Pantalla de modificació dels experiments

La pantalla de modificació de l'experiment, té els seus camps propis de nom, descripció, tipus, etc i a més té la part de posar i treure les casetes que formen part de l'experiment.

Llicències

Durant el projecte hem fet servir programari de tota mena amb diverses llicències. En la següent taula fem un resum de totes les elles:

Programari	Llicència
Eclipse Kepler	Eclipse Public License - v 1.0
Apache HTTP Server BootstrapDateTimePicker	Apache License v2.0
PostgreSQL Django D3.js	BSD 3-Clause License
Python	Python License
PIL	Copyright by Secret Labs
Jquery Twitter Bootstrap	MIT License

Taula 3: Resum de llicències del projecte

Pel codi del projecte hem escollit la llicència GPL v3. El procés de sotmetre el codi a una llicència implica afegir dos elements a cada fitxer: una nota de copyright i una declaració de permís de còpia, precisant que el programa es distribueix sota els termes de la Llicència Pública General GNU. Hem afegit a tots els fitxers de codi els acabat amb .py la següent capçalera:



Il·lustració 35: Logotip GPLv3

```
'''
    Copyright 2014 Eulalia Huguet
    This file is part of Experiments.

    Experiments is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    any later version.

    Experiments is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with Experiments. If not, see <http://www.gnu.org/licenses/>.
'''
```

I al directori arrel el fitxer COPYING.txt amb la llicència completa que podeu trobar a :
<http://www.gnu.org/licenses/gpl.txt>

Fer servir la GPL exigeix que totes les versions millorades que es publiquin siguin també de software lliure. D'aquesta manera si algú publica una versió millorada d'Experiments l'haurà de publicar amb les mateixes condicions.

En tot cas no és obligatori publicar el programa modificat, ni cap part del mateix. Es poden fer versions modificades i fer-les servir en privat, sense que es facin públiques. Només és en el moment de la publicació que la GPL exigeix que es posi a disposició dels usuaris el codi font modificat, sota la GPL també.

Hem de parar especial atenció amb les llibreries Javascript que utilitzem que ja tenen llicència i a l'hora de distribuir-les poden ser incompatibles amb la GPL.

La llicència Apache License v2, que usa BootstrapDateTimePicker és compatible amb la GPL. També ho és la llicència BSD de 3 clàusules de usa la D3.js i també ho és la llicència MIT que usa Bootstrap per tant podem publicar l'obra incloent l'aplicació i les llibreries sota la llicència GLP.

La llista de llicències compatibles e incompatibles amb la GPL v3 la podreu trobar a:
<http://www.gnu.org/licenses/license-list.es.html>

Conclusions

Des del punt de vista tecnològic, l'aplicació “Experiments” és una versió simplificada i funcional dels requeriments plantejats a l'inici del treball. És la versió 1.0: una primera aproximació a la solució final i completa per tractar acuradament totes les dades que es recullen al camp d'experimentació.

La tecnologia escollida ha facilitat molt la feina perquè tota la part d'administració està desenvolupada pel framework. És necessari configurar-lo pel que fa a l'idioma, les opcions de cerca de cada element, com s'ha d'ordenar o quina informació volem mostrar-ne.

S'ha complert l'objectiu de crear una eina que permeti als investigadors organitzar les dades i treure'n resultats de manera ràpida i senzilla.

Personalment, la primera vegada que ens vam trobar amb el director del treball de final de màster, allà al novembre de l'any passat, i em va preguntar en quin llenguatge i framework de desenvolupament m'agradaria fer el TFM no vaig saber què dir. Sí, feia molt temps que no programava, l'última assignatura amb pràctiques al màster va ser el curs 2007-2008, no he desenvolupat cap altre projecte personal en aquests anys fins ara i no he fet cap curset de formació en aquest tema que m'obligués a estudiar altre cop. Per tant, va triar ell i va dir: Django i python, i fes servir stacks.

Vaig pensar que tot allò pertanyia a l'espai sideral i que m'anava molt gran. No sé si va veure la meua cara d'espant, perquè immediatament va arreglar-ho dient que Django era un framework que permet fer aplicacions web rapidíssimament. I amb els stacks no em caldria preocupar de tota la instal·lació dels components: del servidor web, de la base de dades, etc. Doncs tenia raó.

En poc temps ho tenia tot funcionant sense problema. Fantàstic! No obstant això, per escriure una línia de codi i que fes el que jo volia que fes he llegit pàgines i pàgines de documentació. Gran part del treball l'he passat llegint i no programant. He millorat la meua comprensió lectora en anglès. I a més a més he après a programar en python i a defensar-me en javascript, a fer servir un nou framework per desenvolupar aplicacions web: Django i he recordat entre d'altres coses: html, css, jquery, ajax.sql...

Treball Futur

Des de l'anàlisi de requeriments que hem deixat uns blocs del programa per a més endavant, recuperem-los:

R3: gràfiques noves i totalment flexibles. Aquesta funcionalitat està pensada per a tenir un editor de gràfiques similar al que podem trobar en un full de calcul. Serien necessaris molt recursos per fer aquesta funcionalitat. Hauríem de convèncer a l'usuari que la millor opció es que es defineixin els gràfics més habituals i deixar modificar-ne els paràmetres.

R6: cerques de dades creuades i la fer gràfica.

R7: Planificador d'experiments. Es refereix a tenir un calendari on programar els esdeveniments. Seria pràctic que un cop s'introdueixen els experiments, automàticament el sistema poses un event en un calendari durant els dies que es fa l'experiment, de quin tipus és i inclús a la descripció es podrien posar algunes dades si existeixen. El calendari es podria sincronitzar amb qualsevol dispositiu mòbil o tauleta.

R13: càlculs sobre les dades: per exemple aplicar la cal·libració de la sonda a les dades. La cal·libració de les sondes segueixen la següent equació: $y = 1,0748x - 1,2367$, on x és el valor que ha recollit la sonda i y el nou valor calibrat.

Bibliografia

Alchin, Marty (2009). ProDjango: The expert's voice in web development. ISBN-13 (pbk): 978-1-4302-1047-4

@mdo i @fat (2010). Bootstrap [en línia]. [S.I.]: Bootstrap Core Team. [Consultat: 7 octubre 2014]. Disponible a Internet: <http://getbootstrap.com/>

APPCELERATOR, INC. (25 setembre 2014). PyDev [en línia]. [S.I.]: Appcelerator, Inc. [Consultat de 7 octubre de 2014]. Disponible a Internet: <http://pydev.org/>

BOSTOCK, MIKE (2013). D3.js - Data-Driven Documents [en línia]. [S.I.]: Mike Bostock. [Consultat: 8 octubre 2014]. Disponible a Internet: <http://d3js.org/>

CREATIVE COMMONS. Creative Commons [en línia]. [S.I.]: Creative Commons. [Consultat: 12 octubre 2014]. Disponible a Internet: <http://creativecommons.org/>

D3 TIPS AND TRICKS (12 juliol 2014) D3 Tips and Tricks [en línia]. [S.I.]: D3 Tips and Tricks [Consultat: 12 octubre 2014] Disponible a Internet: <http://www.d3noob.org>

DJANGOSNIPPETS. Django snippets: Welcome [en línia]. [S.I.]: Django snippets. [Consultat: 11 octubre 2014]. Disponible a Internet: <http://.djangosnippets.org/>

DJANGO SOFTWARE FOUNDATION (2014). Django: Django documentation [en línia]. [S.I.]: Django Software Foundation. [Consultat: 6 octubre 2014]. Disponible a Internet: <http://docs.djangoproject.com/>

DJANGO SOFTWARE FOUNDATION (2014). Django: The Web framework for perfectionists with deadlines [en línia]. [S.I.]: Django Software Foundation. [Consultat: 6 octubre 2014]. Disponible a Internet: <http://www.djangoproject.com>

FREE SOFTWARE FOUNDATION, INC. (13 Agost 2014). El sistema operatiu GNU [en línia]. Estats Units: Free Software Foundation, Inc. [Consultat: 8 octubre 2014]. Disponible a Internet: <http://www.gnu.org/>

GITHUB, INC. (2014). GitHub – Build software better, together [en línia]. GitHub, Inc. [Consultat: 5 octubre 2014]. Disponible a Internet: <http://github.com>

Holovaty, Adrian and Jacob Kaplan-Moss (2009). The Definitive Guide to django. Web Development Done Right. 2n edition. Estats Units. ISBN 13: 978-1-4302-1936-1

Hourieh, Ayman (2008). Learning Website Development with Django. BIRMINGHAM. ISBN 978-1-847193-35-3

JQUERY FOUNDATION (2014). JQuery Foundation [en línia]. [S.I.]: JQuery Foundation. [Consultat: 6 octubre 2014]. Disponible a Internet: <http://jquery.org/>

OPENSOURCE.ORG. The open source initiative [en línia]. [S.I.] Opensource.org. [Consultat: 8 octubre 2014]. Disponible a Internet: <http://opensource.org/>

POSTGRESQL GLOBAL DEVELOPMENT GROUP (2014). PostgreSQL Documentation [en línia]. [S.I.]: The PostgreSQL Global Development Group. [Consultat: 6 octubre 2014]. Disponible a Internet: <http://www.postgresql.org/docs/>

PYTHON SOFTWARE FOUNDATION (2014, 2 octubre). Overview – Python 2.7.8 documentation [en línia]. [S.I.]: Python Software Foundation. [Consultat: 6 octubre 2014]. Disponible a Internet: <https://docs.python.org/2.7/>

PYTHON SOFTWARE FOUNDATION (2014). PyPI - the Python Package Index [en línia]. [S.I.]: Python Software Foundation. [Consultat: 8 octubre 2014]. Disponible a Internet: <https://pypi.python.org/pypi>

STACK OVERFLOW (2014 octubre 10). Stack Overflow [en línia]. [S.I.]: Stack Overflow. [Consultat: 11 octubre 2014]. Disponible a Internet: <http://stackoverflow.com/>

WIKIPEDIA. Wikipedia [en línia]. [S.I.]: Wikipedia. [Consultat: 12 octubre 2014]. Disponible a Internet: <https://www.wikipedia.org/>